BEING PART OF FREE SOFTWARE; THE OPEN SOURCE WAY

Anoop C S
Drishti 2017
Saturday 18th November 2017

about.me

- Free Software enthusiast and Open Source developer
- Software Engineer at Red Hat
- Upstream contributor to Samba and GlusterFS projects
- Stands for overall computer user freedom as visioned by FSF
- Strong believer of the “Open Source” concept

The origin

- It all began with GNU Project in 1983 by Richard Stallman
- Thirst for a free Unix-like OS which includes a kernel, compilers, editors, mail software, graphical interface, libraries and many others
- Establishment of FSF in 1985
- Developed all major components except one – the kernel
- Started work on GNU Hurd

“Hello everybody out there using minix –

I’m doing a (free) operating system (just a hobby, won’t be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I’d like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I’ve currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I’ll get something practical within a few months, and I’d like to know what features most people would want. Any suggestions are welcome, but I won’t promise I’ll implement them”

Birth of GNU/Linux

- Unix-like kernel developed by Linus Torvalds in 1991
- GNU system(lacks a kernel) + Linux(the kernel itself) = GNU/Linux, the complete OS
- Used by millions of people, typically via GNU/Linux distributions
- For those who are not computer experts – GNOME, a graphical desktop

Distributions all around

- Fedora, Red Hat, CentOS, Debian, Ubuntu, OpenSUSE. .
- All based on Linux
- non-free software started to sneak in
- even in kernel as blobs
- GNU comes out with a list of completely free GNU/Linux distros.
- Free version on linux - Linux-libre

What about licensing?

- Is it required? Yes, of course
- Control on Who uses it? How is it used? How to make money?
- Basically governs the usage, modification or re-distribution of software
- The “copyleft” concept
- Free as in speech, not as in beer
- Right to copy, share, modify and improve creative works of authorship
- General Public License(GPL), Creative Commons Attribution Share Alike License(CC BY-SA)
- GNU released GPLv1 in 1989, GPLv2 in 1991
- Linux 0.12 released with GPLv2 in 1992

Licensing establishments

- FSF(Free Software Foundation) in 1985
- OSI(Open Source Initiative) in 1998
- SFC(Software Freedom Conservancy) in 2006

GPL

- Nobody should be restricted by the software they use
- 4 degrees of freedom:
  - the freedom to use the software for any purpose,
  - the freedom to change the software to suit your needs,
  - the freedom to share the software with your friends and neighbors, and
  - the freedom to share the changes you make.

GPL vs others

- Two categories:
  - Copyleft
    - Strong(GPLv2, GPLv3, AGPLv3)
    - Weak(LGPL, MPL, EPL)
  - Permissive(MIT, BSD, Apache License 2.0)
- Free software = GPL? No, not at all
- Why should I use GPL rather than other free software licenses?
- If you are in doubt, be cautious
  - See if it is OSI-approved
  - See if FSF has classified it as "free"

Licensing tips for developers

based on Introduction to open source licensing by Richard Fontana

- Don't use a nonstandard open source license
- Don't write your own license
- Make sure you comply with conditions of any upstream licenses
- Don't use a license you aren't prepared to "self-comply" with
- When considering what license to use for your project, pay special attention to preferences of your target user and contributor community
- Don't release code without any indication of licensing
- Keep intact all upstream legal notices
- If using a GPL-family license, allow use of later versions

Why do I care?

- Who controls your computer?
- Two possibilities:
  - Users control the program
  - The program controls the users
- Software must respect user freedom
- Privacy matters?
- Remember the 4 degrees of freedom
- Injecting malware and abuse for profit

Free Software and Open Source

- Open source is not an enemy to free software
- Free software uplifts the freedom values along with making the code better
- Open source is a development methodology; free software is a social movement
- Open source doesn't just mean access to the source code
- According to FSF - ‘Free Software is Open Source Software, but Open Source software may not necessarily be Free Software.’ Is it true always?
- There are also licenses the FSF recognizes as 'free' that the OSI has never approved.

Making the code open aka the open source way

- It’s everywhere
- Basis for building up a community
- Some interesting terminologies:
  - Community
  - Tactics
  - Strategy
  - Blogs
  - Leaderless organization
  - Version control
  - Open collaboration tools

Being part of FOSS community

- Different roles:
  - Content writer (writing, editing)
  - Designer (design, usability, interaction)
  - People person (event organization, service/support)
  - Developer (programming, packaging, bug filing)
  - Infrastructure (system administration)
  - Web developer (web app development)
- Choose the right one

The so called “process”

- Deciding on an area to be concentrated
- Picking up a comfortable language
- Searching for a suitable FOSS project
  - Analyze the overall structure
  - Know the adopted license
- Source code management
- Communication means
  - Mailing lists, IRC channels etc.

Look out for opportunities!

- GSoC
- FOSS Meetups
- Hackathon
- Offering help on MLs and IRC
- Talk proposals, Conference CFPs
- Blogging
- Find/Report bugs

FIRST THEY IGNORE YOU,
THE THEY LAUGH AT YOU,
THEN THEY FIGHT YOU,
THEN YOU WIN

At Red Hat. . .

- Our mission:
  - To be the catalyst in communities of customers, contributors, and partners creating better technology the open source way
- Making open source technology work for our enterprise customers
- We are proud to be open source leaders
- Red Hat subscription – kicks out traditional licensing model and vendor lock-in
- Freedom and choice for business needs

Upstream|Downstream @ Red Hat

- Distinction between upstream and downstream
  - Upstream - the community project
  - Downstream - product delivered to customers
- Various examples:
  - Fedora is an upstream for Red Hat Enterprise Linux
  - Openstack is an upstream for RDO
- When the best idea win, customers can’t loose.

Red Hat Development Model

- #1 - Contributes directly to FOSS projects in upstream communities
- #2 - Adds additional engineering efforts to make it enterprise ready:
  - Development & Testing
  - Benchmarking & certifications
- #3 – Service to commercial customers
- Any additional modifications/changes done are also made public

Domains involved

- Linux platforms
- IoT(Internet of Things)
- DevOps(software Development and IT OPerations)
- Mobile
- Big Data and Storage
- Containers
- Virtualization
- Middleware
- Security
- Cloud computing

Upstream communities

- Linux
- glibc/GCC
- KVM
- Samba, NFS-Ganesha
- Docker, Kubernetes, Openstack
- GlusterFS, Ceph
- Fedora, GNOME
- Apache and more . . .

Come forward...Be part of it

- Do not hesitate
- First step - be a user
  - Download and install binary packages, be comfortable with the softare
- Found a bug? Report and fix
  - Switch to source install, learn the code-base and try fixing bugs, take help via MIs, IRC etc
- Lacking a feature? Propose and implement
  - Seek help via introducing yourselves through MLs, IRC channels etc
- Speak up
  - Talk about the cool stuffs you did/doing/proposing. Let the world know so that you get more inputs and suggestions

“SHARING IS GOOD, AND WITH DIGITAL TECHNOLOGY, SHARING IS EASY.”

– Richard Stallman

Thank You

@anoopcs
anoopcs@autistici.org

Q & A